Medical & Biological Engineering & Computing (2024) 62:257–273
https://doi.org/10.1007/s11517-023-02912-0

ORIGINAL ARTICLE

Security vulnerabilities in healthcare: an analysis of medical devices and software

Carlos M. Mejía-Granda[1] · José L. Fernández-Alemán[1] · Juan M. Carrillo-de-Gea[1] · José A. García-Berná[1]

Received: 10 January 2023 / Accepted: 14 August 2023 / Published online: 4 October 2023

Abstract

The integration of IoT in healthcare has introduced vulnerabilities in medical devices and software, posing risks to patient safety and system integrity. This study aims to bridge the research gap and provide valuable insights for addressing healthcare vulnerabilities and their mitigation mechanisms. Software vulnerabilities related to health systems from 2001 to 2022 were collected from the National Vulnerability Database (NVD) systematized by software developed by the researchers and assessed by a medical specialist for their impact on patient well-being. The analysis revealed electronic health records, wireless infusion pumps, endoscope cameras, and radiology information systems as the most vulnerable. In addition, critical vulnerabilities were identified, including poor credential management and hard-coded credentials. The investigation provides some insights into the consequences of vulnerabilities in health software products, projecting future security issues by 2025, offers mitigation suggestions, and highlights trends in attacks on life support and health systems are also provided. The healthcare industry needs significant improvements in protecting medical devices from cyberattacks. Securing communication channels and network schema and adopting secure software practices is necessary. In addition, collaboration, regulatory adherence, and continuous security monitoring are crucial. Industries, researchers, and stakeholders can utilize these findings to enhance security and safeguard patient safety.

Keywords Software security · Vulnerability databases · Descriptive study · Software vulnerabilities in health

1 Introduction

An Internet connection is crucial for information sharing, application updates, and firmware security improvements in the rapidly growing Internet of Things (IoT) field. By 2030, approximately 50 billion IoT devices will be connected to the Internet [1]. In addition, the healthcare sector has embraced IoT (it) wearable devices to enhance patient care and daily life improvements [2]. However, the presence of bugs and vulnerabilities poses (means) a significant risk to the performance and security of these devices. Recently, a substantial increase in vulnerabilities has been recorded, emphasizing the need for robust security measures [3].

Vulnerabilities in computer security refer to weaknesses or errors within a system that attackers can exploit to disrupt its normal functioning [4, 5]. Security vulnerabilities can compromise system operations and endanger accessibility, confidentiality, integrity, and availability [6]. Healthcare businesses have been frequent targets of cyberattacks, with 93 reported incidents between 2013 and 2016 [7]. The resulting healthcare data breaches have been shown to cost an average of $4.35 million by 2022 [8].

Vulnerabilities in networked software applications and devices allow attackers to gain control and carry out illicit activities [9, 10]. Attackers seek to alter the regular operations of systems and devices to achieve persistence, access systems at any time, and propagate the attack to vulnerable adjacent and compatible systems [4, 11]. Security gaps seriously threaten healthcare processes that rely heavily on software automation, such as monitoring blood pressure,

✉ Carlos M. Mejía-Granda
carlosmichael.mejiag@um.es

José L. Fernández-Alemán
aleman@um.es

Juan M. Carrillo-de-Gea
jmcdg1@um.es

José A. García-Berná
josealberto.garcia1@um.es

[1] Department of Informatics and Systems, Faculty of Computer Science, University of Murcia, 30100 Murcia, Spain

electrocardiograms (ECG), oxygen saturation, and body temperature [12]. Additionally, software-assisted breathing systems have played a vital role in managing COVID-19 cases [13].

Malfunctioning software in healthcare settings can pose a significant risk to the patient's well-being. Historical incidents such as the Therac-25 accelerator accidents [14] and the potential reconfiguration of pacemakers highlight the potential dangers [15]. In addition, a machine delivering medication exhibiting erratic behavior can lead to critical availability issues, potentially resulting in patient harm or even death [16].

Security misconfigurations, lack of adherence to Secure Software Development Life Cycle (SSDLC) methodologies, omission of standards, and poor coding practices contribute to security breaches [17–19]. Many healthcare applications run on outdated operating systems, exacerbating cybersecurity issues. Common vulnerabilities in healthcare stem from cryptographic attacks, cybercrime, denial-of-service attacks, injection exploits, malware, privilege escalation, and web security exploits [20].

To fight against vulnerabilities, organizations such as the Open Web Application Security Project (OWASP) and the National Institute of Standards and Technology (NIST) offer valuable resources. OWASP publishes the Top 10 Web Application Security Risks report guiding mitigating vulnerabilities [21]. NIST's National Vulnerability Database (NVD) maintains a comprehensive repository of vulnerabilities, including the Common Vulnerabilities and Exposures (CVE) catalog and the associated Common Weakness Enumeration (CWE) [22, 23].

The Common Vulnerability Scoring System (CVSS) classifies the severity of disclosed vulnerabilities [5]. Nowadays, CVSS has undergone three revisions: (i) v1 in 2004 [24], (ii) v2, which includes the CVSS score metric [25], and (iii) v3.1 was published in 2015 for enhancing the process of establishing vulnerability criticality [26]. On the other hand, CWE is a community-developed project to classify security bugs as a list of common software and hardware weaknesses. In addition, it references weakness identification, mitigation, and prevention efforts [27].

Despite the availability of these resources, no metric exists to measure the criticality of life support and medical devices in people's lives. Additionally, there is a lack of studies focusing on vulnerabilities in healthcare devices and software utilizing official databases like the NVD. This article aims to fill that gap by providing an in-depth analysis of security vulnerabilities in healthcare, specifically targeting medical devices and software from 2001 to 2022, presenting a projection of security issues to rise in 2025 as the cybercriminal tendency and its possible mitigation mechanisms.

2 Related work

Mobile device and healthcare network vulnerabilities, classified by Hasan et al. [28] as integrity and confidentiality, privacy, and availability, have been examined. The audit conducted by the authors concludes that the healthcare network demonstrates the availability by resisting Denial of Service (DoS) attacks, Wi-Fi password cracking, and Address Resolution Protocol (ARP) poisoning attacks. However, protection against Wi-Fi password guessing, ARP poisoning, and reverse engineering is essential to ensure the confidentiality and integrity of medical records. Furthermore, the audit reveals that specific medical devices, including pumping machines, remain susceptible to DoS attacks, despite communication with the monitoring system through an SSL-enabled channel. Moreover, the central Electronic Medical Record (EMR) system and the vital sign monitor system of the inspected hospital transmit sensitive login information over Wi-Fi without utilizing the SSL protocol. Therefore, securing communication channels, network schema, medical devices, and technological equipment is crucial for healthcare sector security.

Abouzakhar et al. [29] assess risks and threats associated with IoT security in the healthcare sector. As healthcare systems increasingly adopt distributed cloud computing schemas, new risks emerge in cloud security. These risks include DoS, unauthorized access, ARP poisoning, VM backdoors, hypervisor attacks, rootkit attacks, and VM escape. The primary source of these dangers stems from the lack of interoperability between different IoT protocols and platforms. Disruptions or corruptions in these systems can lead to significant damage or life-threatening risks. Critical IoT systems must maintain a secure and resilient operating environment to mitigate these threats. Cybersecurity for vital IoT systems, such as healthcare systems, presents a challenging and critical issue.

Farhadi et al. [30] conducted a vulnerability scan on an open-source Electronic Health Record (EHR) and medical practice management software OpenEMR. The scan identifies vulnerabilities such as cross-site scripting, file inclusion, HTTP response splitting, control flow attacks, reflection injection, and encryption and decryption issues related to patient history information. The security breaches identified indicate that OpenEMR does not comply with the Health Insurance Portability and Accountability Act (HIPAA) security requirements. The authors recommend patching vulnerable EHR systems and implementing cryptographic methods for persisting and storing PHI data to address these vulnerabilities.

Martinez [31] focuses on determining metrics and vulnerability gaps in medical devices. Despite the Food and Drug Administration (FDA) oversight of these medical devices, a lack of availability and integrity hinders the creation of a relevant list of failure cases. The author emphasizes the need to analyze and classify each medical device connected to the network based on its risk level and complexity. In addition, security breaches should be stored in a centralized record system to enable organization and mitigation. Compromising the integrity and availability of medical devices, such as pacemakers and insulin pumps, poses genuine risks, potentially resulting in injuries or fatalities.

Zakina McGee et al. [32] perform a security analysis of OpenEMR using vulnerability scanning tools. The vulnerabilities identified include cross-site scripting (XSS), SQL injection, and path traversal attacks. The authors successfully mitigate these vulnerabilities after applying scripts and headers in each PHP code file. Early detection of vulnerabilities is essential, and their severity should be prioritized as high, medium, or low-impact vulnerabilities.

Marquez et al. [33] conduct a Systematic Mapping Study (SMS) to detect, classify, and characterize security vulnerabilities in telehealth systems. The study identifies common security challenges within four classifications: attacks, vulnerabilities, weaknesses, and threats. Four security strategies are highlighted: detecting, stopping/mitigating, reacting, and recovering from attacks. In addition, privacy and unsafe data transmission emerge as critical research topics. Finally, software design, requirements, and models are crucial to create safe telehealth systems.

Tervoort et al. [34] conducted a scoping review to offer remedies for reducing cybersecurity dangers brought on by outdated software in medical equipment. The authors identified eighteen solutions linked to medical devices based on intrusion detection or on providing encrypted communication tunnels after they collected and categorized contributions from a selection of papers. They found that security measures are heavily influenced by the sort of medical device they are protecting.

On the other hand, the use of deep learning to predict vulnerabilities in software has been previously studied in the literature. Some examples can be found in works such as in the literature [35]. This paper surveys and reproduces nine deep-learning models for vulnerability detection on the code. It explores model capabilities, training data, and interpretation, revealing the variability and low agreement among models. In another example [36], the paper discusses cyber vulnerability management in a cybersecurity operations center (CSOC) and proposes a novel framework. This one is Deep VULMAN, which uses deep reinforcement learning and integer programming to prioritize and mitigate vulnerabilities. Results show that the framework outperforms current methods in selecting organization-specific vulnerabilities. The use of artificial intelligence (AI) algorithms to provide information security in distrusted networks is proposed in another paper [37]. Key principles of zero trust are presented, and an architecture based on the service-based architecture (SBA) approach is proposed. Our work does not try to predict vulnerabilities from source or binary code. Our work aims to show an overview of vulnerabilities in healthcare software and their consequences from data collected in the NIST vulnerability database. Recommendations are also provided.

3 Method

The method proposed in this paper encompasses three phases. Firstly, all the software vulnerabilities obtained from NVD are compiled. Secondly, the vulnerabilities related to health and medical systems are filtered by using "HEALTH*" and "MEDIC*" as keyword criteria. This search is supported by a data collection tool that the authors developed. Finally, a descriptive and inferential study was carried out, explicitly taking into account the following metrics: presence, impact, and criticality for health per vulnerability. Figure 1 depicts a research flow chart of the method proposed in this paper.

3.1 Data collection

The NVD site is the chosen repository to collect data, given that its list of vulnerabilities in JavaScript Object Notation (JSON) format is periodically updated. CVE and CWE classify the vulnerabilities. Moreover, information is released using the SCAP specifications [26].

Vulnerabilities from 2001 to 2022 were collected with metrics based on MITRE CVSS version 2.0 because version 3.x did not present all vulnerabilities listed in JSON format. Consequently, the parameters considered were the following:

- **CVE_data_meta: id (ID).** Unique vulnerability identifier.
- **Description: (description_data).** Description of each vulnerability.
- **cvssV2: score (baseScore).** According to the CVSS measure, each vulnerability is given a severity rating from 0 to 10. This metric classifies vulnerabilities as moderate if their values are between 0.0 and 3.9, medium if they fall between 4.0 and 6.9, high between 7.0 and 8.9, and critical if they fall between 9.0 and 10.0 [38].
- **cvssV2: accessComplexity (accessComplexity).** Each vulnerability could have different access levels, such as *high, medium, and low*; complexity metrics are described on the NVD official website.
- **cvssV2: authentication (authentication).** Each vulnerability could have a different authentication level, such

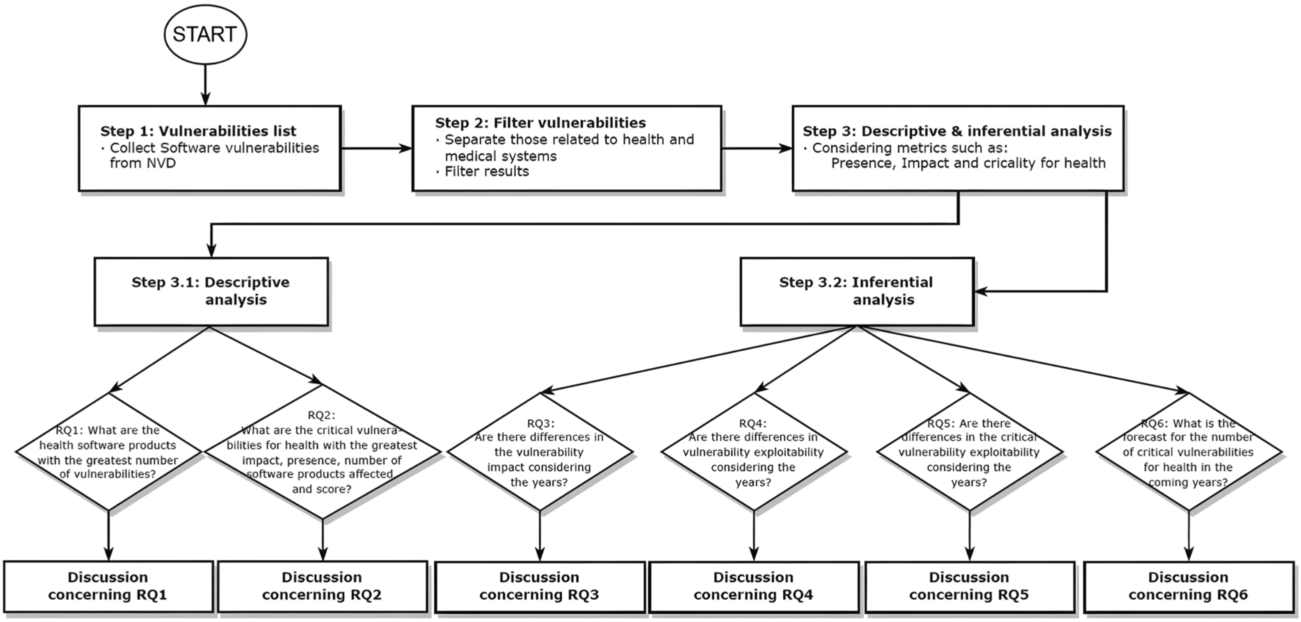

START
Step 1: Vulnerabilities list
· Collect Software vulnerabilities from NVD
Step 2: Filter vulnerabilities
· Separate those related to health and medical systems
· Filter results
Step 3: Descriptive & inferential analysis
· Considering metrics such as: Presence, Impact and cricality for health
Step 3.1: Descriptive analysis
Step 3.2: Inferential analysis
RQ1: What are the health software products with the greatest number of vulnerabilities?
RQ2: What are the critical vulnerabilities for health with the greatest impact, presence, number of software products affected and score?
RQ3: Are there differences in the vulnerability impact considering the years?
RQ4: Are there differences in vulnerability exploitability considering the years?
RQ5: Are there differences in the critical vulnerability exploitability considering the years?
RQ6: What is the forecast for the number of critical vulnerabilities for health in the coming years?
Discussion concerning RQ1
Discussion concerning RQ2
Discussion concerning RQ3
Discussion concerning RQ4
Discussion concerning RQ5
Discussion concerning RQ6

Fig. 1 Research flow chart of the method

as *multiple, single,* or *none*. Authentication metrics are described on the NVD official website.

- **cvssV2: confidentiality Impact (confidentialityImpact).** Each vulnerability has a different confidentiality impact, for instance, *none, partial,* or *complete*. Confidentiality impact metrics are described on the NVD official website.
- **cvssV2: integrity Impact (integrityImpact).** Each vulnerability has a different integrity impact: *none, partial,* or *complete*. Integrity impact metrics are described on the NVD official website.
- **cvssV2: availability Impact (availabilityImpact).** Each vulnerability has a different availability impact metric: *none, partial,* or *complete*. All these metrics are described on the NVD official website.
- **problemtype_data: description (value).** The vulnerabilities are grouped using categories. There are over 1000 vulnerability types, all established per the CWE component of SCAP.
- **baseMetricV2: exploitability (exploitabilityScore).** According to the CVSS measure, each vulnerability has an exploitability level that ranges from 0 to 10. According to this metric, a vulnerability is classified as moderate if its value is between 0.0 and 3.9, medium between 4.0 and 6.9, high between 7.0 and 8.9, and critical between 9.0 and 10.0.

In addition, the **criticality for health** parameters was added. A medical specialist analyzed the description of each vulnerability and its impact. If the software or hardware vulnerability could potentially compromise the patient's life, it will be considered critical for health. Criticality for health metric is expressed by each vulnerability using a binary value: 1 (YES) for critical or 0 (NO) for non-critical matters.

3.2 Obtaining presence and impact per CVE

JSON-formatted data files are annually extracted, which are periodically uploaded and classified by NVD. The calculus of presence metric ($\boldsymbol{Presence_{vuln}}$ Eq. 1) is the number of products affected by a specific vulnerability versus the number of products affected by all vulnerabilities each year. This comparison expresses the relationship between the participation of a vulnerability concerning all vulnerable health software products in the interval. This metric was taken from Calin et al. [39].

$$\mathbf{Presence_{vuln}} = \frac{\mathrm{Products_{vuln}}}{\mathrm{TotalProducts}} \tag{1}$$

The impact value ($\mathbf{Impact_{vuln}}$, Eq. 2) is the relationship between the presence previously calculated and the score found for each vulnerability. This metric was also taken from Calin et al. [39].

$$\mathbf{Impact_{vuln}} = \mathrm{Presence_{vuln}} * \mathrm{Score_{vuln}} \tag{2}$$

The presence and impact calculations are performed for each vulnerability. Furthermore, a JSON file having total presence and impact from NVD-published data feeds is obtained.

3.3 Obtaining presence and impact per each CWE

From collected vulnerabilities, categories are broken down. In this way, a category is associated with its number of vulnerabilities (CVE) and a summary. The average score, which is the average generated from the score meter in vulnerabilities linked to a given category, is the following operation to be carried out.

The presence metric (**Presence**$_{\mathbf{category}}$, Eq. 3) is the ratio of the total number of vulnerabilities detected in each category to those found across all CWE.

$$\mathbf{Presence_{category}} = \frac{\text{Vulnerabilities}_{\text{Category}}}{\text{Total vulnerabilities}_{\text{Categories}}} \tag{3}$$

Finally, the impact metric (**Impact**$_{\mathbf{category}}$, Eq. 4) is the relationship between presence and the average score for each category shown as follows:

$$\mathbf{Impact_{category}} = \text{Presence}_{\text{category}} * \text{Average score}_{\text{category}} \tag{4}$$

Presence and impact calculations must be performed for each category.

3.4 Analyses of trends for the number of critical vulnerabilities in health

To undertake a trend analysis and determine the number of potential future major health vulnerabilities, the least squares method was employed as a root, as applied by García-Berná et al. [40]. The data between the years 2001 and 2022 were therefore used to calculate a straight line (Y, Eq. 5):

$$(1) Y = b \bullet X + a \tag{5}$$

Equation (5) was discovered after solving the following system of equations (Eq. 6, Eq. 7).

$$(2) \sum Y = N \bullet a + b \bullet \sum X \tag{6}$$

$$(3) \sum X \bullet Y = a \bullet \sum X + b \bullet \sum X^2 \tag{7}$$

where N is the number of years studied, Y denotes the number of major vulnerabilities each year, and X represents a conveniently chosen input for the equation to find the coefficients quickly. X values are selected so that their sum equals zero, as shown below (Eq. 8):

$$\sum_{N=2001}^{2022} X_n = 0 \tag{8}$$

3.5 Research questions

Once metrics for categories, vulnerabilities, and trend analysis had been defined, the research questions were formulated. The following questions were answered in this study using descriptive analysis.

- **RQ1.** What are the health software products with the greatest number of vulnerabilities?
- **RQ2.** What are the critical vulnerabilities for health with the greatest impact, presence, number of software products affected, and score?

An inferential analysis will be used to solve the following research questions:

- **RQ3.** Are there differences in the vulnerability impact considering the years?
- **RQ4.** Are there differences in vulnerability exploitability considering the years?
- **RQ5.** Are there differences in the critical vulnerability exploitability considering the years?
- **RQ6.** What is the forecast for the number of critical vulnerabilities for health in the coming years?

IBM SPSS statistical software suite, version 25, was used for data analysis. Before choosing a statistical test over another, some assumptions were considered concerning the data collected in this study. Moreover, when input variables do not follow a normal distribution, a non-parametric test has to be used. Once there are at least three groupings for the variable Years (22 groupings in our study), there are observations of independent studied groups, and the dependent variable has continuous values, the Kruskal–Wallis test is recommended. Additionally, the Kruskal–Wallis H test should be used to compare mean ranks when a researcher analyzes the data morphology and there are abnormal results for several years [41].

4 Data collection tool

To repeat this study, simplify information processing, and contribute to this effort, a Java/Maven processing tool was written with the capability to export information in a CSV spreadsheet. The application is hosted at https://github.com/cmejia5486/nistJson.git.

4.1 JSON data feeds

JSON files must be downloaded from the official NIST website with their respective database organized into years and stored in the ../JsonData directory.

4.1.1 JSON data feed structure

Each JSON file's structure consists of a list of vulnerabilities, where each object's characteristic corresponds to a vulnerability listed in the NVD. A vulnerability object contains its attributes and data collections. The valuable attributes that have been considered for this work are the following.

- **cve: CVE_data_meta:** Contains information related to CVE id per object.
- **cve: Problemtype:** Storage information about the vulnerability's category as defined by the CVE dictionary.
- **cve: Description:** Contains the description of each vulnerability as a summary.
- **Configurations:** Has relevant information related to vulnerable software for each vulnerability.
- **Impact: baseMetricV2:** Contains the CVSS metrics associated with each vulnerability. The most important metrics are cvssV2: accessVector, cvssV2: accessComplexity, cvssV2: authentication, cvssV2: confidentialityImpact, cvssV2: integrityImpact, cvssV2: availabilityImpact, and cvssV2: baseScore
- **Impact: exploitabilityScore:** References to exploitability metric per each vulnerability.

4.1.2 Extra data feed files

The description associated with each category is located in the file ../CweDefinitions/summary.txt. Moreover, there is a file to store criticality parameters for health in ../Health/metrics.csv. In addition, the ../JsonData/Total.json data file was constructed by the union of each data feed file, keeping the JSON structure.

4.2 Source code explanation

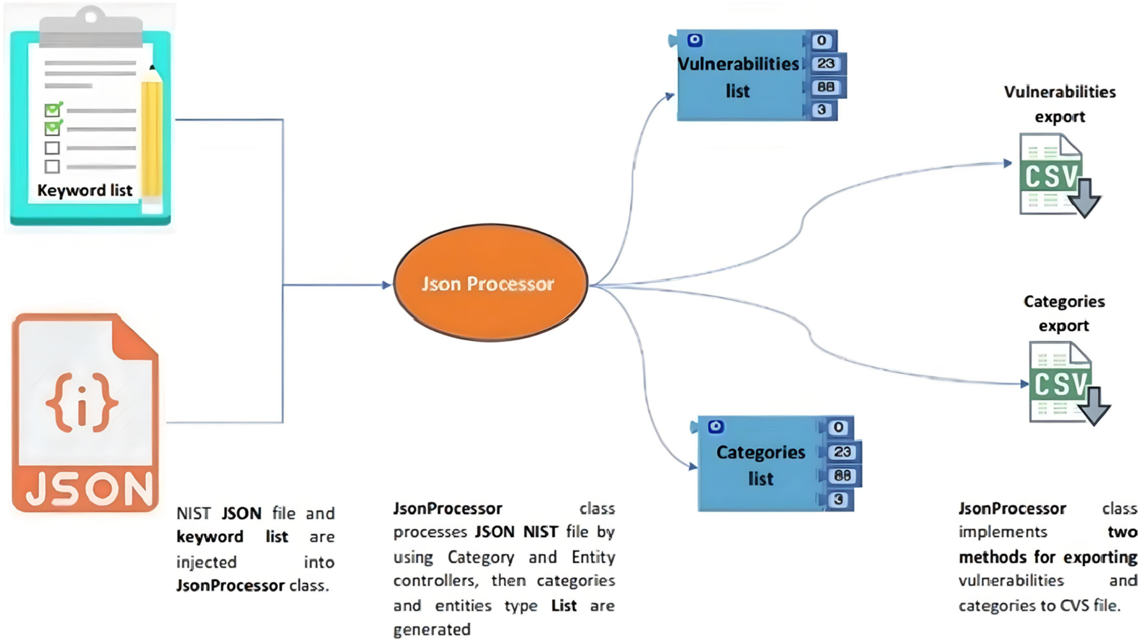
Documentation referring to the Java methods and classes is in the directory ../target/site/apidocs. Figure 2 illustrates the tool's usual execution path. From the JSON file and keywords given for searching, the researcher obtained a list of vulnerabilities, categories, and exclusion vulnerabilities (CVE) after a one-by-one analysis of the vulnerabilities extracted by the tool to exclude false positives. Each vulnerability is analyzed and removed if unrelated to the subject of study. In this work, the file ./Exclusions/exclusions.txt contains 88 CVE items to be excluded, their identifier, and the reason for excluding them from the test according to a medical specialist criterion consequently, the class ../src/nist.main will create summarized CSV files for being analyzed by the researcher with all the specific data presented in Section 3.1 (Fig. 4).

5 Results

In this section, we outline the results of the proposed RQs. Appendices will be included in the repository to provide graphical support for answering each question. The repository is available at https://github.com/cmejia5486/p1_appendixes.git

5.1 Descriptive analysis

Information on vulnerabilities and categories was collected in January 2023. The total number of vulnerabilities related



NIST **JSON** file and **keyword list** are injected into **JsonProcessor** class.

JsonProcessor class processes **JSON NIST** file by using Category and Entity controllers, then categories and entities type **List** are generated

JsonProcessor class implements **two methods for exporting** vulnerabilities and categories to CVS file.

Fig. 2 Data collection tool diagram

Table 1 Top ten health software products with the greatest number of vulnerabilities between 2001 and 2022

N	Software	N. CVE	Vendor
1	librehealth_ehr	21	LibreHealth
2	gehealthcare:centricity	11	General Electric
3	oracle:argus_safety	10	Oracle Corporation
4	artmedic_webdesign:artmedic	9	ArtMedic
5	swisslog-healthcare:hmi-3_control_panel	8	Swisslog
6	smiths-medical:medfusion	8	smiths-medical
7	oracle:industry_applications	7	Oracle Corporation
8	gehealthcare:discovery	6	General Electric
9	medicomp:medcin_engine	5	medicomp
10	ishekar:endoscope_camera	5	Ishekar

to medical vulnerabilities in software and devices was 201 from 2001 to 2022.

5.1.1 Vulnerabilities analysis

> RQ1. What are the health software products with the greatest number of vulnerabilities?

The top ten health software products with the greatest quantity of vulnerabilities are detailed in Table 1. The first fifty software products with the greatest number of vulnerabilities are presented in Appendix A Fig. 4.

> RQ2. What are the critical vulnerabilities for health with the greatest impact, presence, number of software products affected, and score?

There were 106 of the 201 vulnerabilities considered critical for health. The top ten vulnerabilities with the greatest impact are detailed in Table 2. All health vulnerabilities ranked from highest to lowest impact are shown in Appendix B Fig. 5a.

The top ten vulnerabilities with the greatest presence are detailed in Table 3. The complete list of health vulnerabilities ordered from highest to lowest presence is shown in Appendix B Fig. 5b.

The top ten health vulnerabilities grouped by the number of software products affected are detailed in Table 4. All health vulnerabilities ranked from highest to lowest number of software products are shown in Appendix B Fig. 5c.

The top ten health vulnerabilities sorted in order from highest to lowest score are detailed in Table 5. The complete list is shown in Appendix B Fig. 5d.

Criticality for health vulnerabilities is highly influenced by (i) improper management of credentials (user/password), authentication, or incorrect privilege management. For example, the lack of complete proof that an actor's claim of having a particular identity is valid prevents the software from correctly assigning, modifying, tracking, or checking rights for the actor, leading to unexpected consequences and an absence of control; (ii) out-of-bounds reading and writing which causes data corruption, crash, or improper code execution; (iii) hard-coded credentials usually allow an attacker to bypass the authentication and communicate with external devices or systems, even, before performing crucial functionality that needs a user identification that can be verified or that also uses a lot of resources, the software frequently does not do any authentication.

Table 2 Top ten criticality for health vulnerabilities with the greatest impact between 2001 and 2022

N	Vulnerability	Impact	Vendor	Year
1	CVE-2022–22766	0.186	Becton, Dickinson, and Company (BD)	2022
2	CVE-2021–27410	0.138	Hillrom	2021
3	CVE-2018–14786	0.111	Becton, Dickinson, and Company (BD)	2018
4	CVE-2018–4846	0.111	Siemens AG	
5	CVE-2021–27408	0.092	Hillrom	2021
6	CVE-2021–32025	0.080	Blackberry	
7	CVE-2018–4845	0.072	Siemens AG	2018
8	CVE-2017–14006	0.069	GE Healthcare	2017
9	CVE-2016–8355	0.066	Smiths Medical – ICU Medical	2016
10	CVE-2021–22156	0.063	Blackberry	2021

Table 3 Top ten criticality for health vulnerabilities with the greatest presence between 2001 and 2022

N	Vulnerability	Presence	Vendor	Year
1	CVE-2022–22766	0.088	Becton, Dickinson, and Company (BD)	2022
2	CVE-2021–27410	0.018	Hillrom	2021
3	CVE-2021–27408	0.018		
4	CVE-2018–14786	0.015	Becton, Dickinson, and Company (BD)	2018
5	CVE-2018–4846	0.011	Siemens AG	
6	CVE-2021–32025	0.011	Blackberry	2021
7	CVE-2018–4845	0.011	Siemens AG	2018
8	CVE-2017–14006	0.009	GE Healthcare	2017
9	CVE-2021–22156	0.009	Blackberry	2021
10	CVE-2016–8355	0.007	Smiths Medical – ICU Medical	2016

Table 4 Top ten criticality for health vulnerabilities with the greatest number of software products between 2001 and 2022

N	Vulnerability	N. Software	Vendor	Year
1	CVE-2022–22766	48	Becton, Dickinson, and Company (BD)	2022
2	CVE-2021–27410	10	Hillrom	2021
3	CVE-2021–27408	10		2021
4	CVE-2018–14786	8	Becton, Dickinson, and Company (BD)	2018
5	CVE-2018–4846	6	Siemens AG	
6	CVE-2021–32025	6	Blackberry	2021
7	CVE-2018–4845	6	Siemens AG	2018
8	CVE-2017–14006	5	GE Healthcare	2017
9	CVE-2021–22156	5	Blackberry	2021
10	CVE-2016–8355	4	Smiths Medical – ICU Medical	2016

Table 5 Top ten criticality for health vulnerabilities with the greatest score between 2001 and 2022

N	Vulnerability	Score	Vendor	Year
1	CVE-2014–5406	9.3	Hospira	2014
2	CVE-2019–11687	9.3	Nema	2019
3	CVE-2016–8355	9.0	Smiths Medical – ICU Medical	2016
4	CVE-2020–11439	9.0	LibreHealth	2020
5	CVE-2022–31496	9.0		2022
6	CVE-2021–37166	7.8	Swisslog-Healthcare	2021
7	CVE-2021–27410	7.5	Hillrom	
8	CVE-2018–14786	7.5	Becton, Dickinson, and Company (BD)	2018
9	CVE-2017–14006	7.5	GE Healthcare	2017
10	CVE-2017–12726	7.5	Smiths Medical – ICU Medical	

Also, corrupt data, arbitrary code execution, system crash/stoppage due to out-of-confines writing, or improper restriction of operations inside the bounds of a memory buffer are additional effects of the vulnerabilities examined in this study. This type of vulnerability is severe because this study's analyzed devices/software is related to medical support and patient life care.

Conversely, the primary affected industries were as follows:

- GE Healthcare systems, because of sensitive information exposure and poor management of credentials, an attacker can access sensitive personal health information (PHI). Medical information could be used for perpetuating cybercrimes by using false identities with complete names, patient numbers, social security, etc.
- Smiths Medical – ICU Medical software presents a loss of confidentiality and integrity because of incorrect input handling and poor management of credentials. Due to

corrupted information managed by the medication safety software, a patient's medicine can be mistakenly administered and finally threaten their health.

- Hillrom Industries – Welch Allyn medical device management tools are prone to data corruption and malicious code execution due to buffer overflow. Therefore, patient health supported by pressure gauges, visual control devices, cardiac meters, and vital monitoring could be compromised.
- Becton, Dickinson, and Company (BD), with Alaris medical devices product, compromise confidentiality, integrity, and availability due to a weakness related to improper authentication. As a result, patients' health could be critically compromised since a system that controls the supply of the medicine delivered by syringes is undermined.

5.2 Hypothesis test

The data was examined once global data for health/medical scope between 2001 and 2022 regarding the impact, presence, exploitability score, products affected per vulnerabilities, and categories were presented. As a result, some behavior patterns were identified that led us to propose the following research questions:

> RQ3. Are there differences in the vulnerability impact considering the years?

In the statistical analysis, the independent variable was the year, and the dependent variable was the vulnerability impact. After comparing the average ranks of the impact of the vulnerabilities between 2001 and 2022 using the Kruskal–Wallis test, taking into account a mean of 0.05 and a standard deviation of 0.261, statistically significant differences were found ($\chi^2(2)=47.765$, $p=0.001$). In addition, six pairings of years with a p-value under 0.05 showed statistically significant differences in the post hoc contrasts presented in Table 6.

Consequently, the impact metric in 2013 had the highest value, with an average of 0.056, and presented statistically significant differences from the mean of the other years, according to the values shown in Table 6.

Table 6 Statistically significant differences in post hoc contrast for impact between 2001 and 2022

N	Years couple	Mean in years	Contrast statistic	*P*. sig
1	2022–**2017**	0.019–**0.035**	74.204	0.000
2	2022–**2013**	0.019–**0.056**	92.669	0.000
3	2022–**2010**	0.019–**0.033**	83.558	0.002
4	2020–**2017**	0.019–**0.035**	49.979	0.003
5	2020–**2013**	0.019–**0.056**	68.444	0.003
6	2015–**2013**	0.019–**0.056**	75.899	0.004

The years in which a major impact of vulnerabilities was produced is highlighted in boldface

> RQ4. Are there differences in vulnerability exploitability considering the years?

In the statistical analysis, the independent variable was the year and the dependent the vulnerability exploitability. Comparing the mean ranks of the exploitability of the vulnerabilities between 2001 and 2022 using the Kruskal–Wallis test, statistically significant differences were found ($\chi^2(2)=38.461$, $p=0.011$), taking into account a mean of 8.17 and a standard deviation of 2.095. In the post hoc contrasts (see Table 7), statistically significant differences were found in 5 pairs of years (p-value less than 0.05).

Consequently, according to our findings, the exploitability metric in 2010 and 2012 has the highest value with a mean of 10. It has statistically significant differences from the mean of the other years, according to the values shown in Table 7.

> RQ5. Are there differences in the critical vulnerability exploitability considering the years?

In the statistical analysis, the independent variable was the year, and the dependent variable, the critical vulnerability exploitability. A total of 106 out of the 201 software vulnerabilities are critical for health. Applying the Kruskal–Wallis test to compare its mean ranks between 2001 and 2022, statistically significant differences were found ($\chi^2(2)=37.347$, $p=0.007$), taking into account a mean of 8.381 and a standard deviation of 2.175. In the post hoc contrasts (see Table 8), statistically significant variations were discovered in six pairs of years (p-value less than 0.05).

Consequently, the metric of the exploitability vulnerabilities critical for health in 2010 and 2012 have the highest value with an average of 10 and presents statistically

Table 7 Statistically significant differences in exploitability between 2001 and 2022

N	Years couple	Mean in years	Contrast statistic	*P*. sig
1	2014–**2010**	6.988 – **10**	86.353	0.001
2	2020–**2010**	7.457 – **10**	79.435	0.002
3	2016–**2010**	6.067 – **10**	122.333	0.002
4	2013–**2010**	6.711 – **10**	88.056	0.003
5	2016–**2012**	6.067 – **10**	122.333	0.004

The years in which a major impact of vulnerabilities was produced is highlighted in boldface

Table 8 Statistically significant differences for exploitability in vulnerabilities critical for health between 2001 and 2022

N	Years couple	Mean in years	Contrast statistic	*P*. sig
1	2020–**2010**	6.839 – **10.000**	46.538	0.002
2	2016–**2010**	6.067 – **10.000**	63.833	0.002
3	2020–**2021**	6.839 – **9.269**	-34.385	0.002
4	2022–**2010**	7.615– **10.000**	44.615	0.003
5	2022–**2021**	7.615 – **9.269**	32.462	0.004
6	2016–**2012**	6.067 – **10.000**	63.833	0.004

The years in which a major impact of vulnerabilities was produced is highlighted in boldface

significant differences from the mean of the other years, according to the values shown in Table 8.

> RQ6.What is the forecast for the number of critical vulnerabilities for health in the coming years?

After using the least squares method to estimate the number of critical vulnerabilities for health that may emerge in the future (until 2024), the straight line calculated and shown in Table 9 predicts that one more vulnerability will be developed each year compared to the previous one.

Figure 3 shows that the tendency of the number of critical vulnerabilities for health per year versus the actual number per year is expected to grow.

6 Discussion

Results from Section 5 were assessed to determine the study's principal conclusions.

> RQ1. What are the health software products with the greatest number of vulnerabilities?

The study's findings indicate that the health software solutions with the most vulnerabilities are EHR, wireless infusion pumps, endoscope cameras, and radiology information systems. However, despite having security techniques for ensuring medical devices, many vulnerabilities are still present [42, 43]. Considering that health software products support more of the activities in hospitals, several studies agree with us in making it secure

Table 9 Number of critical vulnerabilities for health projected to 2024

Year	The actual number of critical vulnerabilities for health (*Y*)	Input (*X*)	*X*·*Y*	X^2	Number of critical vulnerabilities for health forecast
2025		29			15.007
2024		27			14.320
2023		25			13.634
2022	13	23			12.947
2021	13	21	273	441	12.260
2020	13	19	247	361	11.573
2019	4	17	221	289	10.886
2018	8	15	60	225	10.199
2017	19	13	104	169	9.512
2016	3	11	209	121	8.825
2015	1	9	27	81	8.137
2014	5	7	7	49	7.452
2013	4	5	25	25	6.765
2012	4	3	12	9	6.078
2011	5	1	4	1	5.391
2010	5	−1	−5	1	4.704
2009	1	−3	−15	9	4.017
2008	0	−5	−5	25	3.330
2007	2	−7	0	49	2.644
2006	1	−9	−18	81	1.957
2005	0	−11	−11	121	1.270
2004	1	−13	0	169	0.583
2003	1	−15	−15	225	−0.104
2002	2	−17	−17	289	−0.791
2001	1	−19	−38	361	−1.478

against cyberattacks before use in patients to avoid potential health threats [44, 45].

RQ2. What are the critical vulnerabilities for health with the greatest impact, presence, number of software products affected, and score?

The critical vulnerabilities for health with the greatest impact, presence, number of software products affected, and score are majority caused by (i) poor management of credentials enabling unintended actions and lack of control; (ii) buffer out-of-bounds causing data corruption and improper code execution; and (iii) hard-coded credentials which bypass the authentication in software functions. These weaknesses coincide with the security challenges in health care addressed by Randolph C. Barrows et al. [46]. Unfortunately, encrypted passwords could be present in all programming languages and all operating platforms; this is why it is recommended to mitigate this vulnerability by using [47]:

- *Design (for default accounts):* Use an "initial login" mode that necessitates the user entering a distinct strong password rather than hard-coding a default username and password for first-time logins.
- *Design (for front-end to back-end connections):* There are three potential answers, but none is perfect. **(i) First recommendation** calls for randomly generated passwords that are updated automatically and that a system administrator must enter at predetermined intervals. These passwords will be stored in memory and are only effective during the designated periods. **(ii) Second**, rather than allowing unlimited access, the passwords used should be restricted at the back end to only executing operations required for the front end. **(iii) Finally**, to avoid replay-style assaults, the messages sent should be tagged, verified, and summarized with time-sensitive values.

Moreover, results exposed in RQ2 also revealed that the primary industries involved in criticality for health care are GE Healthcare systems, Smiths Medical – ICU Medical software (syringe infusion pumps), Hillrom industries (Welch Allyn medical device), Becton, Dickinson, and Company (Alaris medical devices), Siemens AG and Blackberry, meaning that health care field continues being through security breaches. For example, syringe infusion pumps are still being attacked by FTP server exploitation [48].

RQ3. Are there differences in the vulnerability impact considering the years?

Despite having an approval and regulatory process for medical devices by the FDA since 1976 [49], the year with the highest impact mean value was 2013. Insulin pumps, X-ray systems, blood refrigeration units, and so on are still vulnerable and are targets for cybercriminals [7]. Patient health information and social security numbers are highly profitable on the dark web, sold 10 to 20 times more than other data types [50, 51]. Moreover, medical data stolen by cyber criminals continue causing negative impacts with numerous losses in patients' treatment plans and hospital operations [52]. Implications for software and healthcare devices can be mitigated by (i) eliciting secure software requirements, (ii) implementing vulnerability detection before the deployment, and (iii) testing and reviewing medical devices during all phases by manufacturers, framed in current regulations [9].

RQ4. Are there differences in vulnerability exploitability considering the years?

The highest exploitability mean was in the years 2010 and 2012. Our findings align with an analysis conducted by the Kenna Security of more than 100,000 vulnerabilities disclosed since 2011 [53]. However, the percentage of exploitable vulnerabilities has dropped over the past years. Exploitability can be due to poor and inadequate infrastructure [54], lack of political will [55], low technology acceptance [56], minimal research [55], limited connectivity [57], inadequate human resources [58, 59], lack of policies and legal framework [60], and using legacy operating systems such as Windows XP [61]. However, the weakest link in cybersecurity and security failures across many information technology domains lies in human errors, which must be minimized and controlled [62]. Other good practices for mitigating exploitability are (i) protecting health information against privileged; (ii) reducing to the minimum the number and type of privileged accounts [63]; (iii) periodically changing management, credential updating, logging, and monitoring according to best ITIL practices [64]; and (iv) implementing ongoing life cycle processes and continued safety post-market monitoring by manufacturers according to FDA [65]. The developers and security analysts must prioritize managing the vulnerabilities based mainly on two criteria: ease of being exploited by attackers and severity of the damage caused to the system.

RQ5. Are there differences in the critical vulnerability exploitability considering the years?

In contrast, with the general trend, the exploitability of critical vulnerabilities is increasing, with a peak of 10 achieved in 2010 and 2012. Considering that vulnerabilities critical to health are potentially dangerous for patients' health, assessing the exploitability of vulnerabilities is an essential concern for security defenders, developers, and manufacturers. However, finding exploitable states in software and devices takes a long time [66, 67]. Forecasting vulnerability exploitability is essential to previous decisions and efforts before being exploited because, unfortunately, it is difficult to patch all vulnerabilities [68]. Providentially, machine learning predicts exploitable vulnerabilities before discovering them [69].

RQ6. What is the forecast for the number of critical vulnerabilities for health in the coming years?

Security gaps and vulnerabilities have been boosted since 2010 [61]. Once holes are published, their exploitability magnitude increases five times [70]. Only in 2019, software vulnerabilities have increased by more than 130.000, according to open-source databases [67]. By our results is expected that at least one vulnerability critical for health increase each year until 2025.

7 Conclusions and further work

This research identified vulnerable industries and studied security breaches affecting electronic devices and software used in critical patient support. It also examined trends and techniques used by cybercriminals to exploit security gaps in eHealth systems. The findings provide recommendations for healthcare software industries, researchers, and users to develop secure software solutions and robust applications and implement security patches or enhancements.

The analysis revealed that the user remains the weakest link in the security chain. There are persistent vulnerabilities, such as poor credential management, sensitive information exposure, and incorrect authentication at well-known companies that produce critical health devices. This upward trend of security breaches in healthcare devices and software is expected to continue.

Implementing recommended security measures and adopting a Secure Software Development Life Cycle (SSDLC) can improve system and device security by mitigating vulnerabilities.

Future work involves the creation of a catalog of safety requirements aligned with regulations, best practices, and standards to ensure quality control in the development of healthcare systems and devices. In addition, a continuous improvement audit method will also be developed to control the vulnerabilities identified during the production stage and before the market launch.

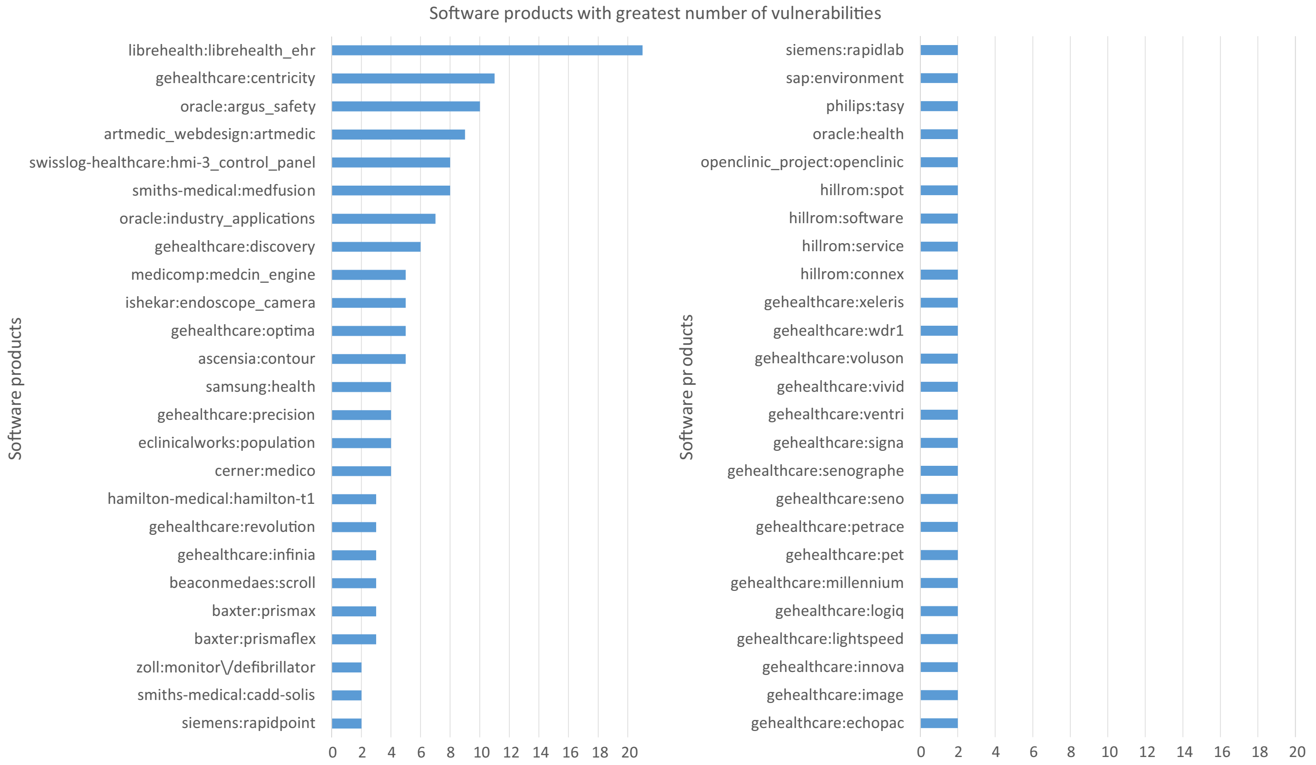


Fig. 3 The first fifty software products with the greatest number of vulnerabilities

Appendix A

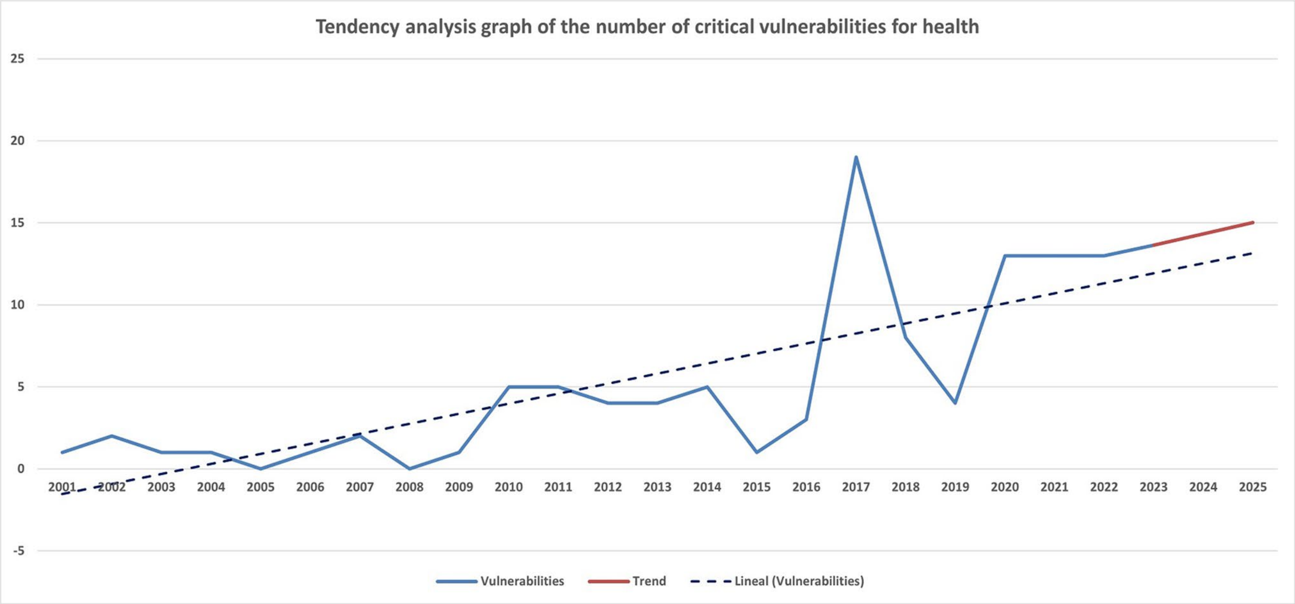


Fig. 4 Tendency analysis graph of the number of critical vulnerabilities for health

Appendix B

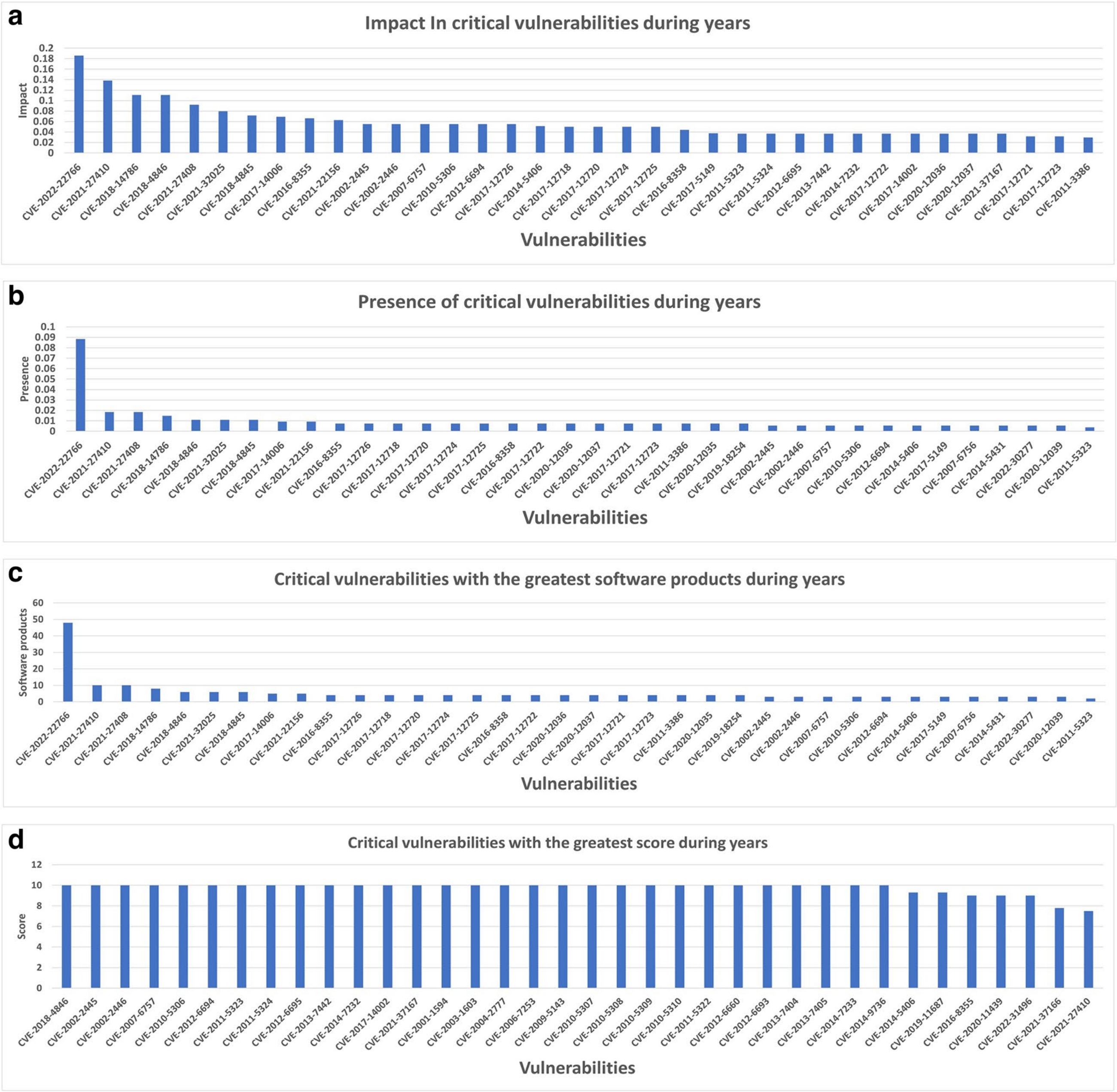


Fig. 5 **a** Health vulnerabilities ranked from highest to lowest impact from 2001 to 2022. **b** Health vulnerabilities ordered from highest to lowest presence from 2001 to 2022. **c** Health vulnerabilities ranked from highest to lowest number of software products from 2001 to 2022. **d** Health vulnerabilities sorted from highest to lowest score from 2001 to 2022

Acknowledgements This research is part of the OASSIS-UMU project (PID2021-122554OB-C32) and the Network of Excellence in Software Quality and Sustainability (RED2022-134656-T), all funded by MCIN/AEI /10.13039/501100011033/ and by ERDF A way to make Europe.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Ethics Approval This article contains no studies with human participants or animals performed by authors.

Conflict of Interest The authors declare no competing interests.

References

1. Statista (2020) Number of connected devices worldwide 2030 | Statista. Statista Research Department. https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/ (accessed Aug. 09, 2021).
2. Xiao F, Miao Q, Xie X, Sun L, Wang R (2018) Indoor anti-collision alarm system based on wearable Internet of Things for smart healthcare. IEEE Commun Mag 56(4):53–59. https://doi.org/10.1109/MCOM.2018.1700706
3. "Browse cve vulnerabilities by date." https://www.cvedetails.com/browse-by-date.php (accessed Mar. 20, 2020)
4. Shamal PK, Rahamathulla K, Akbar A (2018) A study on software vulnerability prediction model, in Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017, Institute of Electrical and Electronics Engineers Inc 703–706. https://doi.org/10.1109/WiSPNET.2017.8299852
5. Ruohonen J (2019) A look at the time delays in CVSS vulnerability scoring. Appl Comput Informa 15(2):129–135. https://doi.org/10.1016/j.aci.2017.12.002
6. Iannone E, Guadagni R, Ferrucci F, De Lucia A, Palomba F (2022) The secret life of software vulnerabilities: a large-scale empirical study. IEEE Trans Software Eng 1. https://doi.org/10.1109/TSE.2022.3140868
7. Beavers J, Pournouri S (2018) Recent cyber attacks and vulnerabilities in medical devices and healthcare institutions BT - blockchain and clinical trial: securing patient data," H. Jahankhani, S. Kendzierskyj, A. Jamal, G. Epiphaniou, and H. Al-Khateeb, Eds., Cham: Springer International Publishing 249–267. https://doi.org/10.1007/978-3-030-11289-9_11
8. "Cost of a data breach report 2021 | IBM." https://www.ibm.com/security/data-breach (accessed Aug. 09, 2021)
9. Lin G, Wen S, Han Q-L, Zhang J, Xiang Y (2020) Software vulnerability detection using deep neural networks: a survey. Proc IEEE 108(10):1825–1848. https://doi.org/10.1109/JPROC.2020.2993293
10. Lin G et al (2018) Cross-project transfer representation learning for vulnerable function discovery. IEEE Trans Industr Inform 14(7):3289–3297. https://doi.org/10.1109/TII.2018.2821768
11. Zhang X, Xie H, Yang H, Shao H, Zhu M (2020) A general framework to understand vulnerabilities in information systems. IEEE Access 8:121858–121873. https://doi.org/10.1109/ACCESS.2020.3006361
12. Islam SMR, Kwak D, Kabir MH, Hossain M, Kwak K-S (2015) The Internet of Things for health care: a comprehensive survey. IEEE Access 3:678–708. https://doi.org/10.1109/ACCESS.2015.2437951
13. Phua J et al. (2020) Intensive care management of coronavirus disease 2019 (COVID-19): challenges and recommendations. The Lancet Respiratory Medicine (8):5. Lancet Publishing Group 506–517. https://doi.org/10.1016/S2213-2600(20)30161-2
14 Burns AJ, Johnson ME, Honeyman P (2016) A brief chronology of medical device security. Commun ACM 59(10):66–72. https://doi.org/10.1145/2890488
15. Goodman M (2011) Who does the autopsy? Criminal implications of implantable medical devices. USENIX Association 4. https://doi.org/10.5555/2028026.2028030
16. Coventry L, Branley D (2018) Cybersecurity in healthcare: a narrative review of trends, threats and ways forward. Maturitas 113:48–52. https://doi.org/10.1016/j.maturitas.2018.04.008
17. Rindell K, Ruohonen J, Holvitie J, Hyrynsalmi S, Leppänen V (2021) "Security in agile software development: a practitioner survey. Inf Softw Technol 131:106488. https://doi.org/10.1016/j.infsof.2020.106488
18. Tung Y, Lo S, Shih J, Lin H (2016) An integrated security testing framework for Secure Software Development Life Cycle," in *2*016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS) 1–4. https://doi.org/10.1109/APNOMS.2016.7737238
19. Serhane A, Raad M, Raad R, Susilo W (2018) PLC code-level vulnerabilities. In 2018 International Conference on Computer and Applications, ICCA 2018, Institute of Electrical and Electronics Engineers Inc. 348–352. https://doi.org/10.1109/COMAPP.2018.8460287
20. Langer SG (2017) Cyber-security issues in healthcare information technology. J Digit Imaging 30(1):117–125. https://doi.org/10.1007/s10278-016-9913-x
21. Poston H (2020) Mapping the OWASP Top Ten to Blockchain. Procedia Comput Sci 177:613–617. https://doi.org/10.1016/j.procs.2020.10.087
22. Li H, Xi R, Zhao L (2015) Study on the distribution of CVSS environmental score. In 2015 IEEE 5th International Conference on Electronics Information and Emergency Communication 122–125. https://doi.org/10.1109/ICEIEC.2015.7284502
23. Meyers BS, Meneely A (2021) An automated post-mortem analysis of vulnerability relationships using natural language word embeddings. Procedia Comput Sci 184:953–958. https://doi.org/10.1016/j.procs.2021.04.018
24. Mell S (2006) Peter and Scarfone, Karen and Romanosky, "Common vulnerability scoring system SIG." IEEE Secur Priv 4(6):85–89. https://doi.org/10.1109/MSP.2006.145
25. Houmb SH, Franqueira VNL, Engum EA (2010) Quantifying security risk level from CVSS estimates of frequency and impact. J Syst Softw 83(9):1622–1634. https://doi.org/10.1016/j.jss.2009.08.023
26. Nowak M, Walkowski M, Sujecki S (2021) Machine learning algorithms for conversion of CVSS base score from 2.0 to 3.x BT - computational science – ICCS 2021," M. Paszynski, D. Kranzlmüller, V. V Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds., Cham: Springer International Publishing. 255–269

27. Howard M (2009) Improving software security by eliminating the CWE top 25 vulnerabilities. IEEE Secur Priv 7(3):68–71. https://doi.org/10.1109/MSP.2009.69
28. Hasan R, Zawoad S, Noor S, Haque MM, Burke D (2016) How secure is the healthcare network from insider attacks? An audit guideline for vulnerability analysis, In Proceedings - International Computer Software and Applications Conference, IEEE Computer Society 417–422. https://doi.org/10.1109/COMPSAC.2016.129
29. Abouzakhar NS, Jones A, Angelopoulou O (2018) Internet of Things security: a review of risks and threats to healthcare sector, In Proceedings - 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017, Institute of Electrical and Electronics Engineers Inc., 373–378. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.62
30. Farhadi M, Haddad H, Shahriar H (2018) Static analysis of HIPPA security requirements in electronic health record applications. In Proceedings - International Computer Software and Applications Conference, IEEE Computer Society 474–479. https://doi.org/10.1109/COMPSAC.2018.10279
31. Martinez JB (2018) Medical device security in the IoT age, In 2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2018, Institute of Electrical and Electronics Engineers Inc. 128–134. https://doi.org/10.1109/UEMCON.2018.8796531
32. McGee Z, Acharya S (2019) Security analysis of OpenEMR, In Proceedings - 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019, Institute of Electrical and Electronics Engineers Inc. 2655–2660. https://doi.org/10.1109/BIBM47256.2019.8983178
33. Marquez G, Astudillo H, Taramasco C (2020) Security in telehealth systems from a software engineering viewpoint: a systematic mapping study. IEEE Access 8:10933–10950. https://doi.org/10.1109/ACCESS.2020.2964988
34. Tervoort T, De Oliveira MT, Pieters W, Van Gelder P, Olabarriaga SD, Marquering H (2020) Solutions for mitigating cybersecurity risks caused by legacy software in medical devices: a scoping review. IEEE Access 8:84352–84361. https://doi.org/10.1109/ACCESS.2020.2984376
35. Napier K, Bhowmik T, Wang S (2023) An empirical study of text-based machine learning models for vulnerability detection. Empir Softw Eng 28(2):38. https://doi.org/10.1007/s10664-022-10276-6
36. Hore S, Shah A, Bastian ND (2023) Deep VULMAN: a deep reinforcement learning-enabled cyber vulnerability management framework". Expert Syst Appl 221:119734. https://doi.org/10.1016/j.eswa.2023.119734
37. Ramezanpour K, Jagannath J (2022) Intelligent zero trust architecture for 5G/6G networks: principles, challenges, and the role of machine learning in the context of O-RAN". Computer Networks 217:109358. https://doi.org/10.1016/j.comnet.2022.109358
38. "NVD - vulnerability metrics." https://nvd.nist.gov/vuln-metrics/cvss (accessed Jul. 11, 2020)
39 Sánchez MC, De Gea JMC, Fernández-Alemán JL, Garcerán J, Toval A (2020) Software vulnerabilities overview: a descriptive study. Tsinghua Sci Technol 25(2):270–280. https://doi.org/10.26599/TST.2019.9010003
40. García-Berná JA et al (2019) Green IT and sustainable technology development: Bibliometric overview. Sustain Dev 27(4):613–636. https://doi.org/10.1002/SD.1927
41. "Kruskal-Wallis H Test in SPSS Statistics | Procedure, output and interpretation of the output using a relevant example." https://statistics.laerd.com/spss-tutorials/kruskal-wallis-h-test-using-spss-statistics.php (accessed Jun. 01, 2022)
42. Kruse CS, Smith B, Vanderlinden H, Nealand A (2017) Security techniques for the electronic health records. J Med Syst 41(8). https://doi.org/10.1007/S10916-017-0778-4
43. O'Brien G et al (2018) Securing electronic health records on mobile devices. Nist Special Publication 1800–1801. https://doi.org/10.6028/NIST.SP.1800-1
44. Rajendraprasad P, Butakov S, Jaafar F (2018) Information security considerations for wireless infusion pumps, Proceedings - 2018 IEEE 18th International Conference on Software Quality, Reliability, and Security Companion, QRS-C 2018 438–442. https://doi.org/10.1109/QRS-C.2018.00081
45. O'Brien G, Edwards S, Littlefield K, McNab N, Wang S, Zheng K (2018) Securing wireless infusion pumps in healthcare delivery organizations. https://doi.org/10.6028/NIST.SP.1800-8
46. Barrows RC, Clayton PD (1996) Privacy, confidentiality, and electronic medical records. J Am Med Inform Assoc 3(2):139–148. https://doi.org/10.1136/JAMIA.1996.96236282
47. "Use of hard-coded password | OWASP Foundation." https://owasp.org/www-community/vulnerabilities/Use_of_hard-coded_password (accessed Jun. 09, 2022)
48. Yaqoob T, Abbas H, Atiquzzaman M (2019) Security vulnerabilities, attacks, countermeasures, and regulations of networked medical devices—a review. IEEE Commun Surv Tutorials 21(4):3723–3768. https://doi.org/10.1109/COMST.2019.2914094
49. Darrow JJ, Avorn J, Kesselheim AS (2021) FDA regulation and approval of medical devices: 1976–2020. JAMA 326(5):420–432. https://doi.org/10.1001/jama.2021.11171
50. Luna R, Rhine E, Myhra M, Sullivan R, Kruse CS (2016) Cyber threats to health information systems: a systematic review. Technol Health Care 24(1):1–9. https://doi.org/10.3233/THC-151102
51. Humer FJC (n.d) Your medical record is worth more to hackers than your credit card | Reuters." https://www.reuters.com/article/us-cybersecurity-hospitals-idUSKCN0HJ21I20140924 (accessed Nov. 10, 2022)
52. Millard WB (2017) Where bits and bytes meet flesh and blood. Ann Emerg Med 70(3):A17–A21. https://doi.org/10.1016/j.annemergmed.2017.07.008
53. "Kenna Security @ 10: a decade of insights | Kenna security research." https://www.kennaresearch.com/a-decade-of-insights/ (accessed Dec. 15, 2022)
54. Azubuike MC, Ehiri JE (1999) Health information systems in developing countries: benefits, problems, and prospects. J R Soc Promot Health 119(3):180–184. https://doi.org/10.1177/146642409911900309
55. Garner P, Kale R, Dickson R, Dans T, Salinas R (1998) Getting research findings into practice: implementing research findings in developing countries. BMJ 317(7157):531–535. https://doi.org/10.1136/bmj.317.7157.531
56. Ajami S, Arab-Chadegani R (2013) Barriers to implement Electronic Health Records (EHRs). Mater Sociomed 25(3):213–215. https://doi.org/10.5455/msm.2013.25.213-215
57. Drury P (2005) The eHealth agenda for developing countries. World Hosp Health Serv 41(4):38–40
58. Rowe AK, de Savigny D, Lanata CF, Victora CG (2005) How can we achieve and maintain high-quality performance of health workers in low-resource settings? Lancet 366(9490):1026–1035. https://doi.org/10.1016/S0140-6736(05)67028-6
59. Kuehn BM (2007) Global shortage of health workers, brain drain stress developing countries. JAMA 298(16):1853–1855. https://doi.org/10.1001/jama.298.16.1853. (United States)
60. Siddiqi S et al (2009) Framework for assessing governance of the health system in developing countries: gateway to good governance. Health Policy 90(1):13–25. https://doi.org/10.1016/j.healthpol.2008.08.005

61. Argaw ST et al (2020) Cybersecurity of hospitals: discussing the challenges and working towards mitigating the risks. BMC Med Inform Decis Mak 20(1):146. https://doi.org/10.1186/s12911-020-01161-7
62. Pycroft L et al (2016) Brainjacking: implant security issues in invasive neuromodulation. World Neurosurg 92:454–462. https://doi.org/10.1016/j.wneu.2016.05.010
63. Haber MJ (2020) Privileged account management implementation BT - privileged attack vectors: building effective cyber-defense strategies to protect organizations," M. J. Haber, Ed., Berkeley, CA: Apress, 335–359. https://doi.org/10.1007/978-1-4842-5914-6_25
64. Oxford: UCSIA ITIL (2017) A guide to change management. https://resources.martechseries.com/mts-whitepapers/itil-change-management-a-beginners-guide.pdf. Accessed 10 Nov 2022
65. FDA, "Software as a medical device (SAMD): clinical evaluation guidance for industry and Food and Drug Administration staff," 2017. https://resources.martechseries.com/mts-whitepapers/itil-change-management-a-beginners-guide.pdf. (accessed Nov. 10, 2022).
66 Wang R et al (2023) Tunter: assessing exploitability of vulnerabilities with taint-guided exploitable states exploration. Comput Secur 124:102995. https://doi.org/10.1016/j.cose.2022.102995
67. Yin J, Tang M, Cao J, Wang H (2020) Apply transfer learning to cybersecurity: predicting exploitability of vulnerabilities by description. Knowl Based Syst 210:106529. https://doi.org/10.1016/j.knosys.2020.106529
68. Tang M, Yin J, Alazab M, Cao J, Luo Y (2021) Modeling of extreme vulnerability disclosure in smart city industrial environments. IEEE Trans Industr Inform 17(6):4150–4158. https://doi.org/10.1109/TII.2020.3022182
69. Bhatt N, Anand A, Yadavalli VSS (2021) Exploitability prediction of software vulnerabilities. Qual Reliab Eng Int 37(2):648–663. https://doi.org/10.1002/qre.2754
70. Bilge L, Dumitra\cs T (2012) Before we knew it: an empirical study of zero-day attacks in the real world, In Proceedings of the 2012 ACM Conference on Computer and Communications Security, in CCS '12. New York, NY, USA: Association for Computing Machinery, 833–844. https://doi.org/10.1145/2382196.2382284

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Carlos M. Mejía-Granda is a Ph.D. candidate in Informatics Engineering at the Faculty of Computer Science, University of Murcia, Murcia, ES-30100, Spain.

José L. Fernández-Alemán is an associate professor at the Software Engineering Research Group, Faculty of Computer Science, University of Murcia, Murcia, ES-30100, Spain.

Juan M. Carrillo-de-Gea is an associate professor with the Software Engineering Research Group, Faculty of Computer Science, University of Murcia, Murcia, ES-30100, Spain.

José A. García-Berná is a researcher at the Software Engineering Research Group, Faculty of Computer Science, University of Murcia, Murcia, ES-30100, Spain.